Package ‘dreamer’

August 20, 2021

Title  Dose Response Models for Bayesian Model Averaging

Version  3.0.0

Description  Fits (longitudinal) dose-response models utilizing a Bayesian model averaging approach as outlined in Gould (2019) <doi:10.1002/bimj.201700211> for both continuous and binary responses. Functions for plotting and calculating various posterior quantities (e.g. posterior mean, quantiles, probability of minimum efficacious dose, etc.) are also implemented. Copyright Eli Lilly and Company (2019).

URL  https://github.com/rich-payne/dreamer

BugReports  https://github.com/rich-payne/dreamer/issues

License  MIT + file LICENSE

Imports  coda, dplyr (>= 1.0.0), ggplot2, graphics, purrr, rootSolve, rjags (>= 4.8), rlang (>= 0.4.5), stats, tidyr (>= 1.0.2), tidyselect (>= 1.1)

Encoding  UTF-8

RoxygenNote  7.1.1

Suggests  testthat (>= 3.0), fs (>= 1.5), knitr, rmarkdown, tibble, spelling

Config/testthat/edition  3

VignetteBuilder  knitr

Language  en-US

NeedsCompilation  no

Author  Richard Payne [aut, cre],
        William Michael Landau [rev],
        Mitch Thomann [rev],
        Eli Lilly and Company [cph]

Maintainer  Richard Payne <paynestatistics@gmail.com>

Repository  CRAN

Date/Publication  2021-08-20 09:50:02 UTC
R topics documented:

diagnostics.................................................................2
dreamerplot.................................................................3
dreamer_data...............................................................6
dreamer_mcmc..............................................................16
dreamer_plot_prior.......................................................17
model.......................................................................19
model_longitudinal........................................................31
plot_comparison..........................................................33
plot_trace.................................................................35
posterior.................................................................36
post_medx.................................................................39
post_perc_effect.........................................................42
pr_eoi......................................................................44
pr_med......................................................................46
pr_medx....................................................................47
summary.dreamer.........................................................49
summary.dreamer_bma...................................................51

Index 53

---

diagnostics Calculate MCMC Diagnostics for Parameters

Description

Calculate MCMC diagnostics for individual parameters.

Usage

diagnostics(x)

Arguments

x MCMC output from a dreamer model.

Value

A tibble listing the Gelman point estimates and upper bounds (obtained from coda::gelman.diag) and effective sample size (obtained from coda::effectiveSize) for each parameter within each model.
Examples

```r
set.seed(888)
data <- dreamer_data_linear(
    n_cohorts = c(20, 20, 20),
    dose = c(0, 3, 10),
    b1 = 1,
    b2 = 3,
    sigma = 5
)
# Bayesian model averaging
output <- dreamer_mcmc(
data = data,
n_adapt = 1e3,
n_burn = 1e3,
n_iter = 1e4,
n_chains = 2,
silent = FALSE,
mod_linear = model_linear(
    mu_b1 = 0,
    sigma_b1 = 1,
    mu_b2 = 0,
    sigma_b2 = 1,
    shape = 1,
    rate = .001,
    w_prior = 1 / 2
),
mod_quad = model_quad(
    mu_b1 = 0,
    sigma_b1 = 1,
    mu_b2 = 0,
    sigma_b2 = 1,
    mu_b3 = 0,
    sigma_b3 = 1,
    shape = 1,
    rate = .001,
    w_prior = 1 / 2
)
)
# for all models
diagnostics(output)
# for a single model
diagnostics(output$mod_quad)
```

---

**dreamerplot**

*Posterior Plot of Bayesian Model Averaging*
Description

Plots the posterior mean and quantiles over the dose range and plots error bars at the observed doses. If the data argument is specified, the observed means at each dose are also plotted.

plot posterior from Bayesian model averaging.

Usage

## S3 method for class 'dreamer'
plot(
  x,
  doses = attr(x, "doses"),
  times = NULL,
  probs = c(0.025, 0.975),
  data = NULL,
  n_smooth = 50,
  predictive = 0,
  width = bar_width(doses),
  reference_dose = NULL,
  ...
)

## S3 method for class 'dreamer_bma'
plot(
  x,
  doses = x$doses,
  times = x$times,
  probs = c(0.025, 0.975),
  data = NULL,
  n_smooth = 200,
  predictive = 0,
  width = bar_width(doses),
  reference_dose = NULL,
  ...
)

Arguments

x output from a call to dreamer_mcmc.
doses a vector of doses at which to plot the dose response curve.
times a vector of the times at which to plot the posterior (for longitudinal models only).
probs quantiles of the posterior to be calculated.
data a dataframe with column names of "dose" and "response" for individual patient data. Optional columns "n" and "sample_var" can be specified if aggregate data is supplied, but it is recommended that patient-level data be supplied where possible for continuous models, as the posterior weights differ if aggregated data is used. For aggregated continuous data, "response" should be the average of "n" subjects with a sample variance of "sample_var". For aggregated binary data,
"response" should be the number of successes, 'n' should be the total number of subjects (the "sample_var" column is irrelevant in binary cases and is ignored).

n_smooth the number of points to calculate the smooth dose response interpolation. Must be sufficiently high to accurately depict the dose response curve.

predictive the size of sample for which to plot posterior predictive intervals for the mean.

width the width of the error bars.

reference_dose the dose at which to adjust the posterior plot. Specifying a dose returns the plot of \text{pr(trt_dose - trt_reference_dose | data)}.

model definitions created using the model creation functions in model. If arguments are named, the names are retained in the return values.

Value

Returns the ggplot object.

Examples

set.seed(888)
data <- dreamer_data_linear(
  n_cohorts = c(20, 20, 20),
  dose = c(0, 3, 10),
  b1 = 1,
  b2 = 3,
  sigma = 5
)

# Bayesian model averaging
output <- dreamer_mcmc(
data = data,
n_adapt = 1e3,
n_burn = 1e3,
n_iter = 1e4,
n_chains = 2,
silent = FALSE,
mod_linear = model_linear(
  mu_b1 = 0,
  sigma_b1 = 1,
  mu_b2 = 0,
  sigma_b2 = 1,
  shape = 1,
  rate = .001,
  w_prior = 1 / 2
),
mod_quad = model_quad(
  mu_b1 = 0,
  sigma_b1 = 1,
  mu_b2 = 0,
  sigma_b2 = 1,
  mu_b3 = 0,
  sigma_b3 = 1,
Generate Data from Dose Response Models

Description

See the model definitions below for specifics for each model.

Usage

dreamer_data_linear(
  n_cohorts,
  doses,
  b1,
  b2,
  sigma,
  times,
  t_max,
  longitudinal = NULL,
  ...
)

dreamer_data_linear_binary(
  n_cohorts,
  doses,
  b1,
  b2,
  link,
  times,
```r
dreamer_data

t_max,
longitudinal = NULL,
...
)
dreamer_data_quad(
n_cohorts,
doses,
b1,
b2,
b3,
sigma,
times,
t_max,
longitudinal = NULL,
...
)
dreamer_data_quad_binary(
n_cohorts,
doses,
b1,
b2,
b3,
link,
times,
t_max,
longitudinal = NULL,
...
)
dreamer_data_loglinear(
n_cohorts,
doses,
b1,
b2,
sigma,
times,
t_max,
longitudinal = NULL,
...
)
dreamer_data_loglinear_binary(
n_cohorts,
doses,
b1,
b2,
dreamer_data

  link,
times,
t_max,
longitudinal = NULL,
...
)
dreamer_data_logquad(
  n_cohorts,
doses,
b1,
b2,
b3,
sigma,
times,
t_max,
longitudinal = NULL,
...
)
dreamer_data_logquad_binary(
  n_cohorts,
doses,
b1,
b2,
b3,
link,
times,
t_max,
longitudinal = NULL,
...
)
dreamer_data_emax(
  n_cohorts,
doses,
b1,
b2,
b3,
b4,
sigma,
times,
t_max,
longitudinal = NULL,
...
)
dreamer_data_emax_binary(
dreamer_data

n_cohorts,
doses,
b1,
b2,
b3,
b4,
link,
times,
t_max,
longitudinal = NULL,
...
)

dreamer_data_exp(
  n_cohorts,
doses,
b1,
b2,
b3,
sigma,
times,
t_max,
longitudinal = NULL,
...
)

dreamer_data_exp_binary(
  n_cohorts,
doses,
b1,
b2,
b3,
link,
times,
t_max,
longitudinal = NULL,
...
)

dreamer_data_beta(
  n_cohorts,
doses,
b1,
b2,
b3,
b4,
scale,
sigma,
times,
t_max,
longitudinal = NULL,
...
)
dreamer_data_beta_binary(
  n_cohorts,
doses,
b1,
b2,
b3,
b4,
scale,
link,
times,
t_max,
longitudinal = NULL,
...
)
dreamer_data_independent(
  n_cohorts,
doses,
b1,
sigma,
times,
t_max,
longitudinal = NULL,
...
)
dreamer_data_independent_binary(
  n_cohorts,
doses,
b1,
link,
times,
t_max,
longitudinal = NULL,
...
)

**Arguments**

- **n_cohorts** a vector listing the size of each cohort.
- **doses** a vector listing the dose for each cohort.
- **b1, b2, b3, b4** parameters in the models. See sections below for each parameter’s interpretation.
in a given model.

sigma     standard deviation.
times     the times at which data should be simulated if a longitudinal model is specified.
t_max     the t_max parameter used in the longitudinal model.
longitudinal a string indicating the longitudinal model to be used. Can be "linear", "itp", or "idp".
...     additional longitudinal parameters.
link     character vector indicating the link function for binary models.
scale     a scaling parameter (fixed, specified by the user) for the beta models.

Value
A dataframe of random subjects from the specified model and parameters.

Functions

- dreamer_data_linear: generate data from linear dose response.
- dreamer_data_linear_binary: generate data from linear binary dose response.
- dreamer_data_quad: generate data from quadratic dose response.
- dreamer_data_quad_binary: generate data from quadratic binary dose response.
- dreamer_data_loglinear: generate data from log-linear dose response.
- dreamer_data_loglinear_binary: generate data from binary log-linear dose response.
- dreamer_data_logquad: generate data from log-quadratic dose response.
- dreamer_data_logquad_binary: generate data from log-quadratic binary dose response.
- dreamer_data_emax: generate data from EMAX dose response.
- dreamer_data_emax_binary: generate data from EMAX binary dose response.
- dreamer_data_exp: generate data from exponential dose response.
- dreamer_data_exp_binary: generate data from exponential binary dose response.
- dreamer_data_beta: generate data from Beta dose response.
- dreamer_data_beta_binary: generate data from binary Beta dose response.
- dreamer_data_independent: generate data from an independent dose response.
- dreamer_data_independent_binary: generate data from an independent dose response.

Linear

\[
y \sim N(f(d), \sigma^2)
\]

\[
f(d) = b_1 + b_2 \ast d
\]

\[
b_1 \sim N(\mu_{b1}, \sigma_{b1})
\]

\[
b_2 \sim N(\mu_{b2}, \sigma_{b2})
\]

\[
1/\sigma^2 \sim \text{Gamma}(\text{shape}, \text{rate})
\]
Quadratic

\[ y \sim N(f(d), \sigma^2) \]
\[ f(d) = b_1 + b_2 \ast d + b_3 \ast d^2 \]
\[ b_1 \sim N(mu_{b1}, sigma_{b1}) \]
\[ b_2 \sim N(mu_{b2}, sigma_{b2}) \]
\[ b_3 \sim N(mu_{b3}, sigma_{b3}) \]
\[ 1/\sigma^2 \sim Gamma(shape, rate) \]

Log-linear

\[ y \sim N(f(d), \sigma^2) \]
\[ f(d) = b_1 + b_2 \ast \log(d + 1) \]
\[ b_1 \sim N(mu_{b1}, sigma_{b1}) \]
\[ b_2 \sim N(mu_{b2}, sigma_{b2}) \]
\[ 1/\sigma^2 \sim Gamma(shape, rate) \]

Log-quadratic

\[ y \sim N(f(d), \sigma^2) \]
\[ f(d) = b_1 + b_2 \ast \log(d + 1) + b_3 \ast \log(d + 1)^2 \]
\[ b_1 \sim N(mu_{b1}, sigma_{b1}) \]
\[ b_2 \sim N(mu_{b2}, sigma_{b2}) \]
\[ b_3 \sim N(mu_{b3}, sigma_{b3}) \]
\[ 1/\sigma^2 \sim Gamma(shape, rate) \]

EMAX

\[ y \sim N(f(d), \sigma^2) \]
\[ f(d) = b_1 + (b_2 - b_1) \ast d_4^b / (\exp(b_3 \ast b_4) + d_4^b) \]
\[ b_1 \sim N(mu_{b1}, sigma_{b1}) \]
\[ b_2 \sim N(mu_{b2}, sigma_{b2}) \]
\[ b_3 \sim N(mu_{b3}, sigma_{b3}) \]
\[ b_4 \sim N(mu_{b4}, sigma_{b4}), (Truncatedabove0) \]
\[ 1/\sigma^2 \sim Gamma(shape, rate) \]

Here, \( b_1 \) is the placebo effect (dose = 0), \( b_2 \) is the maximum treatment effect, \( b_3 \) is the \( \log(ED50) \), and \( b_4 \) is the hill or rate parameter.
Exponential

\[ y \sim N(f(d), \sigma^2) \]
\[ f(d) = b_1 + b_2 \ast (1 - \exp(-b_3 \ast d)) \]
\[ b_1 \sim N(mu_1, sigma_1) \]
\[ b_2 \sim N(mu_2, sigma_2) \]
\[ b_3 \sim N(mu_3, sigma_3), (truncated\ to\ positive) \]
\[ 1/\sigma^2 \sim Gamma(shape, rate) \]

Linear Binary

\[ y \sim Binomial(n, f(d)) \]
\[ link(f(d)) = b_1 + b_2 \ast d \]
\[ b_1 \sim N(mu_1, sigma_1) \]
\[ b_2 \sim N(mu_2, sigma_2) \]

Quadratic Binary

\[ y \sim Binomial(n, f(d)) \]
\[ link(f(d)) = b_1 + b_2 \ast d + b_3 \ast d^2 \]
\[ b_1 \sim N(mu_1, sigma_1) \]
\[ b_2 \sim N(mu_2, sigma_2) \]
\[ b_3 \sim N(mu_3, sigma_3) \]

Log-linear Binary

\[ y \sim Binomial(n, f(d)) \]
\[ link(f(d)) = b_1 + b_2 \ast \log(d + 1) \]
\[ b_1 \sim N(mu_1, sigma_1) \]
\[ b_2 \sim N(mu_2, sigma_2) \]

Log-quadratic Binary

\[ y \sim Binomial(n, f(d)) \]
\[ link(f(d)) = b_1 + b_2 \ast \log(d + 1) + b_3 \ast \log(d + 1)^2 \]
\[ b_1 \sim N(mu_1, sigma_1) \]
\[ b_2 \sim N(mu_2, sigma_2) \]
\[ b_3 \sim N(mu_3, sigma_3) \]
EMAX Binary

\[ y \sim \text{Binomial}(n, f(d)) \]
\[
link(f(d)) = b_1 + (b_2 - b_1) \cdot \frac{d^b}{\exp(b_3 \cdot b_4) + d^b} \\
\]
\[ b_1 \sim N(mu_{b1}, \text{sigma}_{b1}) \]
\[ b_2 \sim N(mu_{b2}, \text{sigma}_{b2}) \]
\[ b_3 \sim N(mu_{b3}, \text{sigma}_{b3}) \]
\[ b_4 \sim N(mu_{b4}, \text{sigma}_{b4}), (\text{Truncated above 0}) \]

Here, on the link(f(d)) scale, \( b_1 \) is the placebo effect (dose = 0), \( b_2 \) is the maximum treatment effect, \( b_3 \) is the log(ED50), and \( b_4 \) is the hill or rate parameter.

Exponential Binary

\[ y \sim \text{Binomial}(n, f(d)) \]
\[
link(f(d)) = b_1 + b_2 \cdot (\exp(b_3 \cdot d) - 1) \\
\]
\[ b_1 \sim N(mu_{b1}, \text{sigma}_{b1}) \]
\[ b_2 \sim N(mu_{b2}, \text{sigma}_{b2}) \]
\[ b_3 \sim N(mu_{b3}, \text{sigma}_{b3}), (\text{Truncated below 0}) \]

Independent

\[ y \sim N(f(d), \sigma^2) \]
\[ f(d) = b_{1d} \]
\[ b_{1d} \sim N(mu_{b1}[d], \text{sigma}_{b1}[d]) \]
\[ 1/\sigma^2 \sim \text{Gamma(shape, rate)} \]

Independent Binary

\[ y \sim \text{Binomial}(n, f(d)) \]
\[
link(f(d)) = b_{1d} \\
\]
\[ b_{1d} \sim N(mu_{b1}[d], \text{sigma}_{b1}[d]) \]

Longitudinal Linear

Let \( f(d) \) be a dose response model. The expected value of the response, \( y \), is:

\[ E(y) = g(d, t) \]
\[ g(d, t) = a + (t/t_{max}) \cdot f(d) \]
\[ a \sim N(mu_a, \text{sigma}_a) \]
Longitudinal ITP

Let \( f(d) \) be a dose response model. The expected value of the response, \( y \), is:

\[
E(y) = g(d, t)
\]

\[
g(d, t) = a + f(d) \cdot \frac{(1 - \exp(-c_1 \cdot t))/((1 - \exp(-c_1 \cdot t_{max})))}{(1 - \exp(-c_1 \cdot t_{max}))}
\]

\[
a \sim N(\mu_a, \sigma_a)
\]

\[
c_1 \sim Uniform(a_{c1}, b_{c1})
\]

Longitudinal IDP

Increasing-Decreasing-Plateau (IDP).

Let \( f(d) \) be a dose response model. The expected value of the response, \( y \), is:

\[
E(y) = g(d, t)
\]

\[
g(d, t) = a + f(d) \cdot I(t < d_1) + (1 - \frac{\exp(-c_2(t - d_1))}{(1 - \exp(-c_2))} \cdot \frac{1}{(1 - \exp(-c_2(t_{max} - d_2)))}) \cdot I(d_1 < t < d_2) + (1 - \frac{\exp(-c_2(t_{max} - d_2))}{(1 - \exp(-c_2))}) \
\]

\[
a \sim N(\mu_a, \sigma_a)
\]

\[
c_1 \sim Uniform(a_{c1}, b_{c1})
\]

\[
c_2 \sim Uniform(a_{c2}, b_{c2})
\]

\[
d_1 \sim Uniform(0, t_{max})
\]

\[
d_2 \sim Uniform(d_1, t_{max})
\]

\[
gam \sim Uniform(0, 1)
\]

Examples

```r
set.seed(888)
data <- dreamer_data_linear(
n_cohorts = c(20, 20, 20),
dose = c(0, 3, 10),
b1 = 1,
b2 = 3,
sigma = 5
)

head(data)

plot(data$dose, data$response)
abline(a = 1, b = 3)
```
dreamer_mcmc  

Bayesian Model Averaging of Dose Response Models

Description

This function performs Bayesian model averaging with a selection of dose response models. See model for all possible models.

Usage

dreamer_mcmc(
  data,
  ...,  
  n_adapt = 1000,  
  n_burn = 1000,  
  n_iter = 10000,  
  n_chains = 4,  
  silent = FALSE,  
  convergence_warn = TRUE
)

Arguments

data  a dataframe with column names of "dose" and "response" for individual patient data. Optional columns "n" and "sample_var" can be specified if aggregate data is supplied, but it is recommended that patient-level data be supplied where possible for continuous models, as the posterior weights differ if aggregated data is used. For aggregated continuous data, "response" should be the average of "n" subjects with a sample variance of "sample_var". For aggregated binary data, "response" should be the number of successes, "n" should be the total number of subjects (the "sample_var" column is irrelevant in binary cases and is ignored).

  ...  model definitions created using the model creation functions in model. If arguments are named, the names are retained in the return values.

n_adapt  the number of MCMC iterations to tune the MCMC algorithm.

n_burn  the number of burn-in MCMC samples.

n_iter  the number of MCMC samples to collect after tuning and burn-in.

n_chains  the number of separate, independent, MCMC chains to run.

silent  logical indicating if MCMC progress bars should be suppressed.

convergence_warn  logical (default TRUE) indicating if the Gelman-Rubin diagnostics should be run to detect convergence issues. Warnings are thrown if the upper bound of the Gelman-Rubin statistic is greater than 1.1.
Details

The Bayesian model averaging approach uses data, multiple models, priors on each model’s parameters, and a prior weight for each model. Using these inputs, each model is fit independently, and the output from the models is used to calculate posterior weights for each model. See Gould (2018) for details.

Value

A named list with S3 class "dreamer_bma" and "dreamer". The list contains the following fields:

- `doses`: a vector of the unique ordered doses in the data.
- `times`: a vector of the unique ordered times in the data.
- `w_prior`: a named vector with the prior probabilities of each model.
- `w_post`: a named vector with the posterior probabilities of each model.
- The individual MCMC fits for each model.

References


dreamer_plot_prior

Description

Plot the prior over the dose range. This is intended to help the user choose appropriate priors.

Usage

dreamer_plot_prior(
  n_samples = 10000,
  probs = c(0.025, 0.975),
  doses,
  n_chains = 1,
  ...,
  times = NULL,
  plot_draws = FALSE,
  alpha = 0.2
)
Arguments

- **n_samples**: the number of MCMC samples per MCMC chain used to generate the plot.
- **probs**: A vector of length 2 indicating the lower and upper percentiles to plot. Not applicable when `plot_draws = TRUE`.
- **doses**: a vector of doses at which to evaluate and interpolate between.
- **n_chains**: the number of MCMC chains.
- **times**: a vector of times at which to plot the prior.
- **plot_draws**: if `TRUE`, the individual draws from the prior are plotted. If `FALSE`, only the prior mean and quantiles are drawn.
- **alpha**: the transparency setting for the prior draws in (0, 1]. Only applies if `plot_draws = TRUE`.

Value

The ggplot object.

Examples

```r
# Plot prior for one model
set.seed(8111)
dreamer_plot_prior(
  doses = c(0, 2.5, 5),
  mod_quad_binary = model_quad_binary(
    mu_b1 = -.5,
    sigma_b1 = .2,
    mu_b2 = -.5,
    sigma_b2 = .2,
    mu_b3 = .5,
    sigma_b3 = .1,
    link = "logit",
    w_prior = 1
  )
)

# plot individual draws
dreamer_plot_prior(
  doses = seq(from = 0, to = 5, length.out = 50),
  n_samples = 100,
  plot_draws = TRUE,
  mod_quad_binary = model_quad_binary(
    mu_b1 = -.5,
    sigma_b1 = .2,
    mu_b2 = -.5,
    sigma_b2 = .2,
    mu_b3 = .5,
    sigma_b3 = .1,
    link = "logit",
    w_prior = 1
  )
)```

```
# plot prior from mixture of models
dreamer_plot_prior(
    doses = c(0, 2.5, 5),
    mod_linear_binary = model_linear_binary(
        mu_b1 = -1,
        sigma_b1 = .1,
        mu_b2 = 1,
        sigma_b2 = .1,
        link = "logit",
        w_prior = .75
    ),
    mod_quad_binary = model_quad_binary(
        mu_b1 = -.5,
        sigma_b1 = .2,
        mu_b2 = -.5,
        sigma_b2 = .2,
        mu_b3 = .5,
        sigma_b3 = .1,
        link = "logit",
        w_prior = .25
    )
)

## Model Creation

### Description

Functions which set the hyperparameters, seeds, and prior weight for each model to be used in Bayesian model averaging via dreamer_mcmc().

See each function’s section below for the model’s details. In the following, $y$ denotes the response variable and $d$ represents the dose.

For the longitudinal specifications, see documentation on model_longitudinal.

### Usage

```r
model_linear(
    mu_b1,
    sigma_b1,
    mu_b2,
    sigma_b2,
    shape,
    rate,
    w_prior = 1,
    longitudinal = NULL
)
```
model_quad(
  mu_b1,
  sigma_b1,
  mu_b2,
  sigma_b2,
  mu_b3,
  sigma_b3,
  shape,
  rate,
  w_prior = 1,
  longitudinal = NULL
)

model_loglinear(
  mu_b1,
  sigma_b1,
  mu_b2,
  sigma_b2,
  shape,
  rate,
  w_prior = 1,
  longitudinal = NULL
)

model_logquad(
  mu_b1,
  sigma_b1,
  mu_b2,
  sigma_b2,
  mu_b3,
  sigma_b3,
  shape,
  rate,
  w_prior = 1,
  longitudinal = NULL
)

model_emax(
  mu_b1,
  sigma_b1,
  mu_b2,
  sigma_b2,
  mu_b3,
  sigma_b3,
  mu_b4,
  sigma_b4,
  shape,
model

rate,
w_prior = 1,
longitudinal = NULL
)

model_exp(
  mu_b1,
sigma_b1,
mu_b2,
sigma_b2,
mu_b3,
sigma_b3,
shape,
rate,
w_prior = 1,
longitudinal = NULL
)

model_beta(
  mu_b1,
sigma_b1,
mu_b2,
sigma_b2,
mu_b3,
sigma_b3,
mu_b4,
sigma_b4,
shape,
rate,
scale = NULL,
w_prior = 1,
longitudinal = NULL
)

model_independent(
  mu_b1,
sigma_b1,
shape,
rate,
doses = NULL,
w_prior = 1,
longitudinal = NULL
)

model_linear_binary(
  mu_b1,
sigma_b1,
mu_b2,
model

model_quad_binary(
  mu_b1,
  sigma_b1,
  mu_b2,
  sigma_b2,
  mu_b3,
  sigma_b3,
  link,
  w_prior = 1,
  longitudinal = NULL
)

model_loglinear_binary(
  mu_b1,
  sigma_b1,
  mu_b2,
  sigma_b2,
  link,
  w_prior = 1,
  longitudinal = NULL
)

model_logquad_binary(
  mu_b1,
  sigma_b1,
  mu_b2,
  sigma_b2,
  mu_b3,
  sigma_b3,
  link,
  w_prior = 1,
  longitudinal = NULL
)

model_emax_binary(
  mu_b1,
  sigma_b1,
  mu_b2,
  sigma_b2,
  mu_b3,
  sigma_b3,
  mu_b4,
model

sigma_b4, 
link, 
w_prior = 1, 
longitudinal = NULL 
)

model_exp_binary( 
mu_b1, 
sigma_b1, 
mu_b2, 
sigma_b2, 
mu_b3, 
sigma_b3, 
link, 
w_prior = 1, 
longitudinal = NULL 
)

model_beta_binary( 
mu_b1, 
sigma_b1, 
mu_b2, 
sigma_b2, 
mu_b3, 
sigma_b3, 
mu_b4, 
sigma_b4, 
scale = NULL, 
link, 
w_prior = 1, 
longitudinal = NULL 
)

model_independent_binary( 
mu_b1, 
sigma_b1, 
doses = NULL, 
link, 
w_prior = 1, 
longitudinal = NULL 
)

Arguments

mu_b1, sigma_b1, mu_b2, sigma_b2, mu_b3, sigma_b3, mu_b4, sigma_b4, shape, rate 
models parameters. See sections below for interpretation in specific models.
w_prior a scalar between 0 and 1 indicating the prior weight of the model.
longitudinal output from a call to one of the model_longitudinal_*() functions. This is used
to specify a longitudinal dose-response model.

scale  a scale parameter in the Beta model. Default is 1.2 * max(dose).

dooses  the doses in the dataset to be modeled. The order of the doses corresponds to
the order in which the priors are specified in mu_b1 and sigma_b1.

link  a character string of either "logit" or "probit" indicating the link function for
binary model.

Value

A named list of the arguments in the function call. The list has S3 classes assigned which are used
internally within dreamer_mcmc().

Linear

\[ y \sim N(f(d), \sigma^2) \]
\[ f(d) = b_1 + b_2 \times d \]
\[ b_1 \sim N(mu_{b1}, sigma_{b1}) \]
\[ b_2 \sim N(mu_{b2}, sigma_{b2}) \]
\[ 1/\sigma^2 \sim Gamma(shape, rate) \]

Quadratic

\[ y \sim N(f(d), \sigma^2) \]
\[ f(d) = b_1 + b_2 \times d + b_3 \times d^2 \]
\[ b_1 \sim N(mu_{b1}, sigma_{b1}) \]
\[ b_2 \sim N(mu_{b2}, sigma_{b2}) \]
\[ b_3 \sim N(mu_{b3}, sigma_{b3}) \]
\[ 1/\sigma^2 \sim Gamma(shape, rate) \]

Log-linear

\[ y \sim N(f(d), \sigma^2) \]
\[ f(d) = b_1 + b_2 \times log(d + 1) \]
\[ b_1 \sim N(mu_{b1}, sigma_{b1}) \]
\[ b_2 \sim N(mu_{b2}, sigma_{b2}) \]
\[ 1/\sigma^2 \sim Gamma(shape, rate) \]
Log-quadratic

\[ y \sim N(f(d), \sigma^2) \]
\[ f(d) = b_1 + b_2 \cdot \log(d + 1) + b_3 \cdot \log(d + 1)^2 \]
\[ b_1 \sim N(mu_{b1}, sigma_{b1}) \]
\[ b_2 \sim N(mu_{b2}, sigma_{b2}) \]
\[ b_3 \sim N(mu_{b3}, sigma_{b3}) \]
\[ 1/\sigma^2 \sim Gamma(shape, rate) \]

EMAX

\[ y \sim N(f(d), \sigma^2) \]
\[ f(d) = b_1 + (b_2 - b_1) \cdot d^3 / (\exp(b_3 \cdot b_4) + d^3) \]
\[ b_1 \sim N(mu_{b1}, sigma_{b1}) \]
\[ b_2 \sim N(mu_{b2}, sigma_{b2}) \]
\[ b_3 \sim N(mu_{b3}, sigma_{b3}) \]
\[ b_4 \sim N(mu_{b4}, sigma_{b4}), (Truncated above 0) \]
\[ 1/\sigma^2 \sim Gamma(shape, rate) \]

Here, \( b_1 \) is the placebo effect (dose = 0), \( b_2 \) is the maximum treatment effect, \( b_3 \) is the \( \log(ED50) \), and \( b_4 \) is the hill or rate parameter.

Exponential

\[ y \sim N(f(d), \sigma^2) \]
\[ f(d) = b_1 + b_2 \cdot (1 - \exp(-b_3 \cdot d)) \]
\[ b_1 \sim N(mu_{b1}, sigma_{b1}) \]
\[ b_2 \sim N(mu_{b2}, sigma_{b2}) \]
\[ b_3 \sim N(mu_{b3}, sigma_{b3}), (truncated to be positive) \]
\[ 1/\sigma^2 \sim Gamma(shape, rate) \]

Beta

\[ y \sim N(f(d), \sigma^2) \]
\[ f(d) = b_1 + b_2 \cdot ((b_3 + b_4)(b_3 + b_4)) / (b_3^4 b_4 + b_4^4) \cdot (d/scale)^b_3 \cdot (1 - d/scale)^b_4 \]
\[ b_1 \sim N(mu_{b1}, sigma_{b1}) \]
\[ b_2 \sim N(mu_{b2}, sigma_{b2}) \]
\[ b_3 \sim N(mu_{b3}, sigma_{b3}), Truncated above 0 \]
\[ b_4 \sim N(mu_{b4}, sigma_{b4}), Truncated above 0 \]
\[ 1/\sigma^2 \sim Gamma(shape, rate) \]

Note that \( scale \) is a hyperparameter specified by the user.
Independent

\[ y \sim N(f(d), \sigma^2) \]
\[ f(d) = b_{1d} \]
\[ b_{1d} \sim N(\mu_{b1}[d], \sigma_{b1}[d]) \]
\[ 1/\sigma^2 \sim Gamma(shape, rate) \]

Independent Details

The independent model models the effect of each dose independently. Vectors can be supplied to \( \mu_{b1} \) and \( \sigma_{b1} \) to set a different prior for each dose in the order the doses are supplied to \( doses \). If scalars are supplied to \( \mu_{b1} \) and \( \sigma_{b1} \), then the same prior is used for each dose, and the \( doses \) argument is not needed.

Linear Binary

\[ y \sim Binomial(n, f(d)) \]
\[ link(f(d)) = b_1 + b_2 \times d \]
\[ b_1 \sim N(\mu_{b11}, \sigma_{b11}) \]
\[ b_2 \sim N(\mu_{b12}, \sigma_{b12}) \]

Quadratic Binary

\[ y \sim Binomial(n, f(d)) \]
\[ link(f(d)) = b_1 + b_2 \times d + b_3 \times d^2 \]
\[ b_1 \sim N(\mu_{b21}, \sigma_{b21}) \]
\[ b_2 \sim N(\mu_{b22}, \sigma_{b22}) \]
\[ b_3 \sim N(\mu_{b23}, \sigma_{b23}) \]

Log-linear Binary

\[ y \sim Binomial(n, f(d)) \]
\[ link(f(d)) = b_1 + b_2 \times log(d + 1) \]
\[ b_1 \sim N(\mu_{b31}, \sigma_{b31}) \]
\[ b_2 \sim N(\mu_{b32}, \sigma_{b32}) \]
Log-quadratic Binary

\[ y \sim \text{Binomial}(n, f(d)) \]
\[ \text{link}(f(d)) = b_1 + b_2 \ast \log(d + 1) + b_3 \ast \log(d + 1)^2 \]
\[ b_1 \sim N(\mu_{b1}, \sigma_{b1}) \]
\[ b_2 \sim N(\mu_{b2}, \sigma_{b2}) \]
\[ b_3 \sim N(\mu_{b3}, \sigma_{b3}) \]

EMAX Binary

\[ y \sim \text{Binomial}(n, f(d)) \]
\[ \text{link}(f(d)) = b_1 + (b_2 - b_1) \ast \left(\frac{d}{d + \exp(b_3 + b_4)}\right) \]
\[ b_1 \sim N(\mu_{b1}, \sigma_{b1}) \]
\[ b_2 \sim N(\mu_{b2}, \sigma_{b2}) \]
\[ b_3 \sim N(\mu_{b3}, \sigma_{b3}) \]
\[ b_4 \sim N(\mu_{b4}, \sigma_{b4}), (\text{Truncated above} 0) \]

Here, on the link(f(d)) scale, \( b_1 \) is the placebo effect (dose = 0), \( b_2 \) is the maximum treatment effect, \( b_3 \) is the \( \log(ED50) \), and \( b_4 \) is the hill or rate parameter.

Exponential Binary

\[ y \sim \text{Binomial}(n, f(d)) \]
\[ \text{link}(f(d)) = b_1 + b_2 \ast (\exp(b_3 \ast d) - 1) \]
\[ b_1 \sim N(\mu_{b1}, \sigma_{b1}) \]
\[ b_2 \sim N(\mu_{b2}, \sigma_{b2}) \]
\[ b_3 \sim N(\mu_{b3}, \sigma_{b3}), (\text{Truncated below} 0) \]

Beta Binary

\[ y \sim \text{Binomial}(n, f(d)) \]
\[ \text{link}(f(d)) = b_1 + b_2 \ast \left(\frac{(b_3 + b_4)(b_3 + b_4)}{(b_3 + b_4)^3 \ast b_4^b \ast (d / \text{scale})^b \ast (1 - d / \text{scale})^b_4} \right) \]
\[ b_1 \sim N(\mu_{b1}, \sigma_{b1}) \]
\[ b_2 \sim N(\mu_{b2}, \sigma_{b2}) \]
\[ b_3 \sim N(\mu_{b3}, \sigma_{b3}), \text{Truncated above} 0 \]
\[ b_4 \sim N(\mu_{b4}, \sigma_{b4}), \text{Truncated above} 0 \]

Note that scale is a hyperparameter specified by the user.
Independent Binary

\[ y \sim Binomial(n, f(d)) \]
\[ \text{link}(f(d)) = b_{1d} \]
\[ b_{1d} \sim N(\mu_{b1}[d], \sigma_{b1}[d]) \]

Independent Binary Details

The independent model models the effect of each dose independently. Vectors can be supplied to \( \mu_{b1} \) and \( \sigma_{b1} \) to set a different prior for each dose in the order the doses are supplied to \( \text{doses} \). If scalars are supplied to \( \mu_{b1} \) and \( \sigma_{b1} \), then the same prior is used for each dose, and the \( \text{doses} \) argument is not needed.

Longitudinal Linear

Let \( f(d) \) be a dose response model. The expected value of the response, \( y \), is:

\[ E(y) = g(d, t) \]
\[ g(d, t) = a + (t/t_{max}) \times f(d) \]
\[ a \sim N(\mu_a, \sigma_a) \]

Longitudinal ITP

Let \( f(d) \) be a dose response model. The expected value of the response, \( y \), is:

\[ E(y) = g(d, t) \]
\[ g(d, t) = a + f(d) \times ((1 - \exp(-c1 \times t))/(1 - \exp(-c1 \times t_{max}))) \]
\[ a \sim N(\mu_a, \sigma_a) \]
\[ c1 \sim Uniform(a_{c1}, b_{c1}) \]

Longitudinal IDP

Increasing-Decreasing-Plateau (IDP).

Let \( f(d) \) be a dose response model. The expected value of the response, \( y \), is:

\[ E(y) = g(d, t) \]
\[ g(d, t) = a + f(d) \times (((1 - \exp(-c1 \times t)))/(1 - \exp(-c1 \times d1)) \times I(t < d1) + (1 - \text{gam} \times ((1 - \exp(-c2 \times (t-d1)))/(1 - \exp(-c2 \times (d2-d1)))) \times I(d1 \leq t < d2) + (1 - \text{gam}) \times I(t \geq d2)) \]
\[ a \sim N(\mu_a, \sigma_a) \]
\[ c1 \sim Uniform(a_{c1}, b_{c1}) \]
\[ c2 \sim Uniform(a_{c2}, b_{c2}) \]
\[ d1 \sim Uniform(0, t_{max}) \]
\[ d2 \sim Uniform(d1, t_{max}) \]
\[ \text{gam} \sim Uniform(0, 1) \]
Examples

```r
set.seed(888)
data <- dreamer_data_linear(
n_cohorts = c(20, 20, 20),
dose = c(0, 3, 10),
b1 = 1,
b2 = 3,
sigma = 5
)

# Bayesian model averaging
output <- dreamer_mcmc(
data = data,
n_adapt = 1e3,
n_burn = 1e2,
n_iter = 1e3,
n_chains = 2,
silent = TRUE,
mod_linear = model_linear(
mu_b1 = 0,
sigma_b1 = 1,
mu_b2 = 0,
sigma_b2 = 1,
shape = 1,
rate = .001,
w_prior = 1 / 2
),
mod_quad = model_quad(
mu_b1 = 0,
sigma_b1 = 1,
mu_b2 = 0,
sigma_b2 = 1,
mu_b3 = 0,
sigma_b3 = 1,
shape = 1,
rate = .001,
w_prior = 1 / 2
)
)
# posterior weights
output$w_post
# plot posterior dose response
plot(output)

# LONGITUDINAL
library(ggplot2)
set.seed(889)
data_long <- dreamer_data_linear(
n_cohorts = c(10, 10, 10, 10), # number of subjects in each cohort
doses = c(.25, .5, .75, 1.5), # dose administered to each cohort
b1 = 0, # intercept
b2 = 2, # slope
)}
model

\[
\begin{align*}
\text{sigma} &= .5, \quad \text{# standard deviation,} \\
\text{longitudinal} &= \text{"itp"}, \\
\text{times} &= \text{c}(0, 12, 24, 52), \\
\text{t_max} &= 52, \quad \text{# maximum time} \\
a &= .5, \\
c1 &= .1
\end{align*}
\]

## Not run:
\[
\text{ggplot(data_long, aes(time, response, group = dose, color = factor(dose))) + geom_point()}
\]

## End(Not run)

\[
\begin{align*}
\text{output_long} &\leftarrow \text{dreamer_mcmc(} \\
\text{data} &= \text{data_long,} \\
\text{n_adapt} &= 1e3, \\
\text{n_burn} &= 1e2, \\
\text{n_iter} &= 1e3, \\
\text{n_chains} &= 2, \\
\text{silent} &= \text{TRUE}, \quad \text{# make rjags be quiet,} \\
\text{mod_linear} &= \text{model_linear(} \\
\text{mu_b1} &= 0, \\
\text{sigma_b1} &= 1, \\
\text{mu_b2} &= 0, \\
\text{sigma_b2} &= 1, \\
\text{shape} &= 1, \\
\text{rate} &= .001, \\
\text{w_prior} &= 1 / 2, \quad \text{# prior probability of the model} \\
\text{longitudinal} &= \text{model_longitudinal_itp(} \\
\text{mu_a} &= 0, \\
\text{sigma_a} &= 1, \\
a_c1 &= 0, \\
b_c1 &= 1, \\
t_max &= 52
\text{)}
\text{)} \\
\text{mod_quad} &= \text{model_quad(} \\
\text{mu_b1} &= 0, \\
\text{sigma_b1} &= 1, \\
\text{mu_b2} &= 0, \\
\text{sigma_b2} &= 1, \\
\text{mu_b3} &= 0, \\
\text{sigma_b3} &= 1, \\
\text{shape} &= 1, \\
\text{rate} &= .001, \\
\text{w_prior} &= 1 / 2, \\
\text{longitudinal} &= \text{model_longitudinal_linear(} \\
\text{mu_a} &= 0, \\
\text{sigma_a} &= 1, \\
t_max &= 52
\text{)}
\text{)}
\end{align*}
\]
## Not run:

```r
# plot longitudinal dose-response profile
plot(output_long, data = data_long)
plot(output_long$mod_quad, data = data_long) # single model

# plot dose response at final timepoint
plot(output_long, data = data_long, times = 52)
plot(output_long$mod_quad, data = data_long, times = 52) # single model

## End(Not run)
```

---

### Description

Assign hyperparameters and other values for longitudinal modeling. The output of this function is intended to be used as the input to the `longitudinal` argument of the dose response model functions, e.g., `model_linear`.

### Usage

```r
model_longitudinal_linear(mu_a, sigma_a, t_max)

model_longitudinal_itp(mu_a, sigma_a, a_c1 = 0, b_c1 = 1, t_max)

model_longitudinal_idp(
  mu_a,
  sigma_a,
  a_c1 = 0,
  b_c1 = 1,
  a_c2 = -1,
  b_c2 = 0,
  t_max
)
```

### Arguments

- `mu_a`, `sigma_a`, `a_c1`, `b_c1`, `a_c2`, `b_c2`
  - hyperparameters of the specified longitudinal model. See below for parameterization.

- `t_max`
  - a scalar, typically indicating the latest observed time for subjects. This will influence the interpretation of the parameters of each model.
Value

A named list of the arguments in the function call. The list has S3 classes assigned which are used internally within dreamer_mcmc().

Longitudinal Linear

Let \( f(d) \) be a dose response model. The expected value of the response, \( y \), is:

\[
E(y) = g(d, t) \\
g(d, t) = a + (t/t_{max}) * f(d) \\
a \sim N(\mu_a, \sigma_a)
\]

Longitudinal ITP

Let \( f(d) \) be a dose response model. The expected value of the response, \( y \), is:

\[
E(y) = g(d, t) \\
g(d, t) = a + f(d) * ((1 - exp(-c1 * t))/(1 - exp(-c1 * t_{max}))) \\
a \sim N(\mu_a, \sigma_a) \\
c1 \sim Uniform(a_{c1}, b_{c1})
\]

Longitudinal IDP

Increasing-Decreasing-Plateau (IDP).

Let \( f(d) \) be a dose response model. The expected value of the response, \( y \), is:

\[
E(y) = g(d, t) \\
g(d, t) = a + f(d) * (((1-exp(-c1*t))/(1-exp(-c1*d1)))*I(t < d1)+ (1-gam*((1-exp(-c2*(t-d1))))/(1-exp(-c2*tm))/((1-exp(-c2*tm))))/((1-exp(-c2*tm)))/((1-exp(-c2*tm)))) \\
a \sim N(\mu_a, \sigma_a) \\
c1 \sim Uniform(a_{c1}, b_{c1}) \\
c2 \sim Uniform(a_{c2}, b_{c2}) \\
d1 \sim Uniform(0, t_{max}) \\
d2 \sim Uniform(d1, t_{max}) \\
gam \sim Uniform(0, 1)
\]
Description

Compare Posterior Fits

Usage

plot_comparison(..., doses, times, probs, data, n_smooth, width)

## Default S3 method:
plot_comparison(
    ...
    doses = attr(list(...)[[1]], "doses"),
    times = NULL,
    probs = c(0.025, 0.975),
    data = NULL,
    n_smooth = 50,
    width = bar_width(doses)
)

## S3 method for class 'dreamer_bma'
plot_comparison(
    ...
    doses = x$doses,
    times = NULL,
    probs = c(0.025, 0.975),
    data = NULL,
    n_smooth = 50,
    width = bar_width(doses)
)

Arguments

... dreamer_mcmc objects to be used for plotting.
doses a vector of doses at which to plot the dose response curve.
times the times at which to do the comparison.
probs quantiles of the posterior to be calculated.
data a dataframe with column names of "dose" and "response" for individual patient data. Optional columns "n" and "sample_var" can be specified if aggregate data is supplied, but it is recommended that patient-level data be supplied where possible for continuous models, as the posterior weights differ if aggregated data is used. For aggregated continuous data, "response" should be the average of "n" subjects with a sample variance of "sample_var". For aggregated binary data, "response" should be the number of successes, "n" should be the total number of subjects (the "sample_var" column is irrelevant in binary cases and is ignored).
n_smooth: the number of points to calculate the smooth dose response interpolation. Must be sufficiently high to accurately depict the dose response curve.

width: the width of the error bars.

Details

If a Bayesian model averaging object is supplied first, all individual fits and the Bayesian model averaging fit will be plotted, with the model averaging fit in black (other model colors specified in the legend). Otherwise, named arguments must be supplied for each model, and only the models provided will be plotted.

Value

a ggplot object.

Examples

```r
set.seed(888)
data <- dreamer_data_linear(
  n_cohorts = c(20, 20, 20),
  dose = c(0, 3, 10),
  b1 = 1,
  b2 = 3,
  sigma = 5
)

# Bayesian model averaging
output <- dreamer_mcmc(
data = data,
n_adapt = 1e3,
n_burn = 1e3,
n_iter = 1e4,
n_chains = 2,
silent = FALSE,
mod_linear = model_linear(
  mu_b1 = 0,
  sigma_b1 = 1,
  mu_b2 = 0,
  sigma_b2 = 1,
  shape = 1,
  rate = .001,
  w_prior = 1 / 2
),
mod_quad = model_quad(
  mu_b1 = 0,
  sigma_b1 = 1,
  mu_b2 = 0,
  sigma_b2 = 1,
  mu_b3 = 0,
  sigma_b3 = 1,
  shape = 1,
  rate = .001,
)
```r
w_prior = 1 / 2
)
)

plot_comparison(output)

# compare individual models
plot_comparison(linear = output$mod_linear, quad = output$mod_quad)
```

---

**plot_trace**  
*Traceplots*

**Description**

Produces traceplots for each parameter for each model.

**Usage**

```r
plot_trace(x)
```

**Arguments**

- `x`  
  output from a call to `dreamer_mcmc()`.

**Value**

No return value, called to create plots.

**Examples**

```r
set.seed(888)
data <- dreamer_data_linear(
  n_cohorts = c(20, 20, 20),
  dose = c(0, 3, 10),
  b1 = 1,
  b2 = 3,
  sigma = 5
)

# Bayesian model averaging
output <- dreamer_mcmc(
  data = data,
  n_adapt = 1e3,
  n_burn = 1e3,
  n_iter = 1e4,
  n_chains = 2,
  silent = FALSE,
  mod_linear = model_linear(  
    mu_b1 = 0,
    sigma_b1 = 1,
  )
)```
mu_b2 = 0,
sigma_b2 = 1,
shape = 1,
rate = .001,
w_prior = 1 / 2
),
mod_quad = model_quad(
  mu_b1 = 0,
  sigma_b1 = 1,
  mu_b2 = 0,
  sigma_b2 = 1,
  mu_b3 = 0,
  sigma_b3 = 1,
  shape = 1,
  rate = .001,
  w_prior = 1 / 2
)
)

# all parameters from all models
plot_trace(output)

# from a single model
plot_trace(output$mod_linear)

---

**posterior**

*Posterior Quantities from Bayesian Model Averaging*

**Description**

Calculate posterior mean (and quantiles for specific doses for each MCMC iteration of the model.

**Usage**

```r
posterior(
  x,
  doses,
  times,
  probs,
  reference_dose,
  predictive,
  return_samples,
  iter,
  return_stats
)
```

```
## S3 method for class 'dreamer'
posterior(
  x,
```
posterior

doses = attr(x, "doses"),
times = attr(x, "times"),
probs = c(0.025, 0.975),
reference_dose = NULL,
predictive = 0,
return_samples = FALSE,
iter = NULL,
return_stats = TRUE
)

## S3 method for class 'dreamer_mcmc_independent'
posterior(
  x,
  doses = attr(x, "doses"),
times = attr(x, "times"),
  probs = c(0.025, 0.975),
  reference_dose = NULL,
predictive = 0,
  return_samples = FALSE,
  iter = NULL,
  return_stats = TRUE
)

## S3 method for class 'dreamer_bma'
posterior(
  x,
  doses = x$doses,
times = x$times,
  probs = c(0.025, 0.975),
  reference_dose = NULL,
predictive = 0,
  return_samples = FALSE,
  iter = NULL,
  return_stats = TRUE
)

Arguments

x output from a call to dreamer_mcmc.
doses doses at which to estimate posterior quantities.
times a vector of times at which to calculate the posterior response (for longitudinal models only).
probs quantiles of the posterior to be calculated.
reference_dose the dose at which to adjust the posterior plot. Specifying a dose returns the plot of p(rtr_dose - trt_reference_dose I data).
predictive An integer. If greater than 0, the return values will be from the predictive distribution of the mean of predictive observations. If 0 (default), the posterior on the dose response mean is returned.
return_samples: logical indicating if the weighted raw MCMC samples from the Bayesian model averaging used to calculate the mean and quantiles should be returned.

iter: an index on which iterations of the MCMC should be used in the calculations. By default, all MCMC iterations are used.

return_stats: logical indicating whether or not the posterior statistics should be calculated.

Value

A named list with the following elements:

- stats: a tibble the dose, posterior mean, and posterior quantiles.
- samps: the weighted posterior samples. Only returned if return_samples = TRUE.

Methods (by class)

- dreamer: posterior summary for linear model.
- dreamer_mcmc_independent: posterior summary for independent model.
- dreamer_bma: posterior summary for Bayesian model averaging fit.

Examples

```r
set.seed(888)
data <- dreamer_data_linear(
  n_cohorts = c(20, 20, 20),
  dose = c(0, 3, 10),
  b1 = 1,
  b2 = 3,
  sigma = 5
)

# Bayesian model averaging
output <- dreamer_mcmc(
data = data,
n_adapt = 1e3,
n_burn = 1e3,
n_iter = 1e4,
n_chains = 2,
silent = FALSE,
mod_linear = model_linear(
  mu_b1 = 0,
  sigma_b1 = 1,
  mu_b2 = 0,
  sigma_b2 = 1,
  shape = 1,
  rate = .001,
w_prior = 1 / 2
),
mod_quad = model_quad(
  mu_b1 = 0,
  sigma_b1 = 1,
)```

mu_b2 = 0,
sigma_b2 = 1,
mu_b3 = 0,
sigma_b3 = 1,
shape = 1,
rate = .001,
w_prior = 1 / 2
)

posterior(output)

# return posterior samples of the mean
post <- posterior(output, return_samples = TRUE)
head(post$samps)

# from a single model
posterior(output$mod_quad)

# posterior of difference of doses
posterior(output, reference_dose = 0)

---

**post_medx**  
*Posterior Distribution of Minimum X% Effective Dose*

**Description**

Posterior Distribution of Minimum X% Effective Dose

**Usage**

```r
post_medx(  
  x,
  ed,
  probs,
  time,
  lower,
  upper,
  greater,
  small_bound,
  return_samples,
  ...
)
```

```r
## S3 method for class 'dreamer_bma'
post_medx(  
  x,
  ed,
  probs = c(0.025, 0.975),
)
post_medx

time = NULL,
lower = min(x$doses),
upper = max(x$doses),
greater = TRUE,
small_bound = 0,
return_samples = FALSE,
...
)

## S3 method for class 'dreamer'
post_medx(
  x,
  ed,
  probs = c(0.025, 0.975),
time = NULL,
lower = min(attr(x, "doses")),
upper = max(attr(x, "doses")),
greater = TRUE,
small_bound = 0,
return_samples = FALSE,
index = 1:(nrow(x[[1]]) * length(x)),
...
)

Arguments

- **x**: output from `dreamer_mcmc()`.
- **ed**: a number between 0 and 100 indicating the ed% dose that is being sought.
- **probs**: a vector of quantiles to calculate on the posterior.
- **time**: the slice of time for which to calculate the posterior EDX dose. Applies to longitudinal models only.
- **lower**: the lower bound of the doses for calculating EDX.
- **upper**: the upper bound of the doses for calculating EDX.
- **greater**: if TRUE, higher values indicate better efficacy. If FALSE, lower responses indicate better efficacy.
- **small_bound**: the minimum (greater = TRUE) or maximum (greater = FALSE) bound of the response.
- **return_samples**: logical indicating if the posterior samples should be returned.
- **...**: additional arguments for specific methods.
- **index**: a vector indicating which MCMC samples to use in the calculation. If NULL (default), all MCMC samples are used.

Details

The minimum X% effective dose is the dose that has X% of the largest effect for doses between lower and upper. When greater is TRUE, larger positive responses are considered more effective and vice versa. The X% response is calculated as small_bound + ed / 100 * (max_effect -
small_bound) where "max_effect" is the maximum response for doses between lower and upper. The X% effective dose is the smallest dose which has X% response within the dose range. It is possible that for some MCMC samples, an X% effective dose may not exist, so probabilities are not guaranteed to sum to one.

**Value**

Posterior quantities and samples (if applicable), generally in the form of a list. The pr edx exists column gives the posterior probability that an EDX% effect exists.

**Examples**

```r
set.seed(888)
data <- dreamer_data_linear(
n_cohorts = c(20, 20, 20),
dose = c(0, 3, 10),
b1 = 1,
b2 = 3,
sigma = 5
)

# Bayesian model averaging
output <- dreamer_mcmc(
data = data,
n_adapt = 1e3,
n_burn = 1e3,
n_iter = 1e4,
n_chains = 2,
silent = FALSE,
mod_linear = model_linear(
  mu_b1 = 0,
sigma_b1 = 1,
mu_b2 = 0,
sigma_b2 = 1,
shape = 1,
rate = .001,
w_prior = 1 / 2
),
mod_quad = model_quad(
  mu_b1 = 0,
sigma_b1 = 1,
mu_b2 = 0,
sigma_b2 = 1,
mu_b3 = 0,
sigma_b3 = 1,
shape = 1,
rate = .001,
w_prior = 1 / 2
)
)

post_medx(output, ed = c(50, 90))
```
Calculate Posterior of a Dose's Percentage Effect

Description

Given a dose, the "percentage effect" is defined as (effect of the given dose - small_bound) / (maximum effect in dose range - small_bound). This function returns the posterior statistics and/or samples of this effect.

Usage

```r
post_perc_effect(
  x,
  dose,
  probs,
  time,
  lower,
  upper,
  greater,
  small_bound,
  index,
  return_samples
)
```

```
## S3 method for class 'dreamer_bma'
post_perc_effect(
  x,
  dose,
  probs = c(0.025, 0.975),
  time = NULL,
  lower = min(x$doses),
  upper = max(x$doses),
  greater = TRUE,
  small_bound = 0,
  index = NA,
  return_samples = FALSE
)
```

```
## S3 method for class 'dreamer'
post_perc_effect(
  x,
  dose,
  probs = c(0.025, 0.975),
  ```
Arguments

- **x**: output from a call to `dreamer_mcmc()`, or the MCMC samples from a single model of output from a `dreamer_mcmc()` call.
- **dose**: the dose at which to calculate the posterior percentage effect.
- **probs**: a vector of quantiles to calculate on the posterior.
- **time**: the slice of time for which to calculate the posterior percentage effect. Applies to longitudinal models only.
- **lower**: the lower bound of the dose range under consideration.
- **upper**: the upper bound of the dose range under consideration.
- **greater**: logical indicating if the response is desired to be increasing (TRUE) or decreasing (FALSE).
- **small_bound**: the lower (if `greater = TRUE`) or upper (if `greater = FALSE`) bound that the effect is expected to take.
- **index**: an index on which MCMC samples should be used. Generally the user should not specify anything for this argument as `dreamer` will handle this automatically.
- **return_samples**: logical indicating if the posterior samples should be returned.

Value

A named list with the following components:

- **stats**: a tibble listing the dose, time (where relevant), probability a percentage effect exists, the average percentage effect, and the specified quantiles of the percentage effect.
- **samps**: a tibble with the posterior samples for each dose/time combination.

Examples

```r
set.seed(888)
data <- dreamer_data_linear(
  n_cohorts = c(20, 20, 20),
  dose = c(0, 3, 10),
b1 = 1,
b2 = 3,
sigma = 5
)

# Bayesian model averaging
```
output <- dreamer_mcmc(
  data = data,
  n_adapt = 1e3,
  n_burn = 1e3,
  n_iter = 1e4,
  n_chains = 2,
  silent = FALSE,
  mod_linear = model_linear(
    mu_b1 = 0,
    sigma_b1 = 1,
    mu_b2 = 0,
    sigma_b2 = 1,
    shape = 1,
    rate = .001,
    w_prior = 1 / 2
  ),
  mod_quad = model_quad(
    mu_b1 = 0,
    sigma_b1 = 1,
    mu_b2 = 0,
    sigma_b2 = 1,
    mu_b3 = 0,
    sigma_b3 = 1,
    shape = 1,
    rate = .001,
    w_prior = 1 / 2
  )
)
post_perc_effect(output, dose = c(3, 5))

# from a single model
post_perc_effect(output$mod_linear, dose = c(3, 5))

---

**pr_eoi**  
*Calculate Probability of Meeting Effect of Interest (EOI)*

**Description**

Calculate Pr(effect_dose - effect_reference_dose > EOI | data) or Pr(effect_dose > EOI | data).

**Usage**

```r
pr_eoi(x, eoi, dose, reference_dose = NULL, time = NULL)
```

**Arguments**

- `x`  
  output from a call to dreamer_mcmc().

- `eoi`  
  a vector of the effects of interest (EOI) in the probability function.
dose a vector of the doses for which to calculate the posterior probabilities.
reference_dose a vector of doses for relative effects of interest.
time the time at which to calculate the posterior quantity. Defaults to the latest time-point. Applies to longitudinal models only.

Value
A tibble listing the doses, times, and Pr(effect_dose - effect_reference_dose > eoi) if reference_dose is specified; otherwise, Pr(effect_dose > eoi).

Examples
```r
set.seed(888)
data <- dreamer_data_linear(
n_cohorts = c(20, 20, 20),
  dose = c(0, 3, 10),
  b1 = 1,
  b2 = 3,
  sigma = 5
)

# Bayesian model averaging
output <- dreamer_mcmc(
data = data,
n_adapt = 1e3,
n_burn = 1e3,
n_iter = 1e4,
n_chains = 2,
silent = FALSE,
mod_linear = model_linear(
  mu_b1 = 0,
  sigma_b1 = 1,
  mu_b2 = 0,
  sigma_b2 = 1,
  shape = 1,
  rate = .001,
  w_prior = 1 / 2
),
mod_quad = model_quad(
  mu_b1 = 0,
  sigma_b1 = 1,
  mu_b2 = 0,
  sigma_b2 = 1,
  mu_b3 = 0,
  sigma_b3 = 1,
  shape = 1,
  rate = .001,
  w_prior = 1 / 2
)
)
pr_eoi(output, dose = 3, eoi = 10)
```
# difference of two doses
pr_eoi(output, dose = 3, eoi = 10, reference_dose = 0)

# single model
pr_eoi(output$mod_linear, dose = 3, eoi = 10)

---

**Description**

Calculates the posterior probability that each specified doses are the minimum effective dose in the set; i.e. the smallest dose that has a clinically significant difference (CSD).

**Usage**

```r
df <- pr_med(
  x, 
  doses = attr(x, "doses"), 
  csd = NULL, 
  reference_dose = NULL, 
  greater = TRUE, 
  time = NULL
)
```

**Arguments**

- `x`: output from a call to `dreamer_mcmc()`.
- `doses`: the doses for which `Pr(MED)` is to be calculated.
- `csd`: the treatment effect that is clinically relevant.
- `reference_dose`: a single dose that is used as the reference when defining the MED relative to a dose (rather than in absolute terms). When `reference_dose` is specified, this function calculates the posterior probability that each dose is the smallest dose such that `(effect_dose - effect_reference_dose > CSD)`.
- `greater`: if `TRUE`, higher responses indicate better efficacy. If `FALSE`, lower responses indicate better efficacy.
- `time`: the time (scalar) at which the `Pr(MED)` should be calculated. Applies only to longitudinal models.

**Value**

A tibble listing each dose and the posterior probability that each dose is the minimum efficacious dose.
Examples

```r
set.seed(888)
data <- dreamer_data_linear(
  n_cohorts = c(20, 20, 20),
  dose = c(0, 3, 10),
  b1 = 1,
  b2 = 3,
  sigma = 5
)

# Bayesian model averaging
output <- dreamer_mcmc(
data = data,
n_adapt = 1e3,
n_burn = 1e3,
n_iter = 1e4,
n_chains = 2,
silent = FALSE,
mod_linear = model_linear(
  mu_b1 = 0,
  sigma_b1 = 1,
  mu_b2 = 0,
  sigma_b2 = 1,
  shape = 1,
  rate = .001,
  w_prior = 1 / 2
),
mod_quad = model_quad(
  mu_b1 = 0,
  sigma_b1 = 1,
  mu_b2 = 0,
  sigma_b2 = 1,
  mu_b3 = 0,
  sigma_b3 = 1,
  shape = 1,
  rate = .001,
  w_prior = 1 / 2
)
)

pr_med(output, csd = 10)

# difference of two doses
pr_med(output, csd = 3, reference_dose = 0)

# single model
pr_med(output$mod_quad, csd = 10)
```

---

**pr_medx**  
*Probability of minimum X% effective dose*
Description

Calculate the probability a dose being the smallest dose that has at least X% of the maximum efficacy.

Usage

pr_medx(
  x,
  doses = attr(x, "doses"),
  ed,
  greater = TRUE,
  small_bound = 0,
  time = NULL
)

Arguments

x output from a call to dreamer_mcmc().
doses the doses for which pr(minimum effective X% dose) is to be calculated.
ed a number between 0 and 100 indicating the ed% dose that is being sought.
greater if TRUE, higher responses indicate better efficacy. If FALSE, lower responses indicate better efficacy.
small_bound the lower (upper) bound of the response variable when greater = TRUE (FALSE). This is used to calculate the ed% effect as ed / 100 * (effect_100 - small_bound) + small_bound.
time the time (scalar) at which the Pr(MEDX) should be calculated.

Details

Obtaining the probability of a particular does being the minimum efficacious dose achieving ed% efficacy is dependent on the doses specified.

For a given MCMC sample of parameters, the 100% efficacy value is defined as the highest efficacy of the doses specified. For each posterior draw of MCMC parameters, the minimum ed% efficacious dose is defined as the lowest dose what has at least ed% efficacy relative to the 100% efficacy value.

The ed% effect is calculated as ed / 100 * (effect_100 - small_bound) + small_bound where effect_100 is the largest mean response among doses for a given MCMC iteration.

Value

A data frame with the following columns:

- dose: numeric dose levels.
- prob: Prob(EDX | data) for each dose. Note: these probabilities do not necessarily sum to 1 because the EDX may not exist. In fact, Pr(EDX does not exist | data) = 1 - sum(prob).
Examples

```r
set.seed(888)
data <- dreamer_data_linear(
  n_cohorts = c(20, 20, 20),
dose = c(0, 3, 10),
b1 = 1,
b2 = .1,
sigma = 5
)

# Bayesian model averaging
output <- dreamer_mcmc(
data = data,
n_adapt = 1e3,
n_burn = 1e3,
n_iter = 1e4,
n_chains = 2,
silent = FALSE,
mod_linear = model_linear(
  mu_b1 = 0,
sigma_b1 = 1,
mu_b2 = 0,
sigma_b2 = 1,
shape = 1,
rate = .001,
w_prior = 1 / 2
),
mod_quad = model_quad(
  mu_b1 = 0,
sigma_b1 = 1,
mu_b2 = 0,
sigma_b2 = 1,
mu_b3 = 0,
sigma_b3 = 1,
shape = 1,
rate = .001,
w_prior = 1 / 2
)
)
pr_medx(output, ed = 80)

# single model
pr_medx(output$mod_linear, ed = 80)
```

---

**summary.dreamer**

**Summarize Model Output**

**Description**

Produces summaries for inference and diagnosing MCMC chains.
Usage

```r
## S3 method for class 'dreamer'
summary(object, ...)
```

Arguments

- `object`: MCMC output from a dreamer model.
- `...`: additional arguments which are ignored.

Value

A tibble with inference and diagnostics information for each parameter.

Examples

```r
set.seed(888)
data <- dreamer_data_linear(
  n_cohorts = c(20, 20, 20),
  dose = c(0, 3, 10),
  b1 = 1,
  b2 = 3,
  sigma = 5
)

# Bayesian model averaging
output <- dreamer_mcmc(
  data = data,
  n_adapt = 1e3,
  n_burn = 1e3,
  n_iter = 1e4,
  n_chains = 2,
  silent = FALSE,
  mod_linear = model_linear(
    mu_b1 = 0,
    sigma_b1 = 1,
    mu_b2 = 0,
    sigma_b2 = 1,
    shape = 1,
    rate = .001,
    w_prior = 1 / 2
  ),
  mod_quad = model_quad(
    mu_b1 = 0,
    sigma_b1 = 1,
    mu_b2 = 0,
    sigma_b2 = 1,
    mu_b3 = 0,
    sigma_b3 = 1,
    shape = 1,
    rate = .001,
    w_prior = 1 / 2
  )
)
```
summary.dreamer_bma

)

# all models (also show model weights)
summary(output)

# single model
summary(output$mod_linear)

summary.dreamer_bma  Summarize Bayesian Model Averaging MCMC Output

Description

Summarize parameter inference and convergence diagnostics.

Usage

## S3 method for class 'dreamer_bma'
summary(object, ...)

Arguments

object  a dreamer MCMC object.
...
additional arguments (which are ignored).

Value

Returns a named list with elements model_weights and summary containing the prior and posterior weights for each model and inference on parameters for each model as well as MCMC diagnostics.

Examples

set.seed(888)
data <- dreamer_data_linear(
  ncohorts = c(20, 20, 20),
dose = c(0, 3, 10),
b1 = 1,
b2 = 3,
sigma = 5
)

# Bayesian model averaging
output <- dreamer_mcmc(
data = data,
n_adapt = 1e3,
n_burn = 1e3,
n_iter = 1e4,
n_chains = 2,
silent = FALSE,
mod_linear = model_linear(
  mu_b1 = 0,
  sigma_b1 = 1,
  mu_b2 = 0,
  sigma_b2 = 1,
  shape = 1,
  rate = .001,
  w_prior = 1 / 2
),
mod_quad = model_quad(
  mu_b1 = 0,
  sigma_b1 = 1,
  mu_b2 = 0,
  sigma_b2 = 1,
  mu_b3 = 0,
  sigma_b3 = 1,
  shape = 1,
  rate = .001,
  w_prior = 1 / 2
)
)

# all models (also show model weights)
summary(output)

# single model
summary(output$mod_linear)
Index

diagnostics, 2
dreamer_data, 6
dreamer_data_beta(dreamer_data), 6
dreamer_data_beta_binary
  (dreamer_data), 6
dreamer_data_emax(dreamer_data), 6
dreamer_data_emax_binary
  (dreamer_data), 6
dreamer_data_exp(dreamer_data), 6
dreamer_data_exp_binary(dreamer_data), 6
dreamer_data_independent
  (dreamer_data), 6
dreamer_data_independent_binary
  (dreamer_data), 6
dreamer_data_linear(dreamer_data), 6
dreamer_data_linear_binary
  (dreamer_data), 6
dreamer_data_loglinear(dreamer_data), 6
dreamer_data_loglinear_binary
  (dreamer_data), 6
dreamer_data_logquad(dreamer_data), 6
dreamer_data_logquad_binary
  (dreamer_data), 6
dreamer_data_quad(dreamer_data), 6
dreamer_data_quad_binary
  (dreamer_data), 6
dreamer_mcmc, 4, 16, 37
dreamer_plot_prior, 17
dreamerplot, 3

model, 5, 16, 18, 19
model_beta(model), 19
model_beta_binary(model), 19
model_emax(model), 19
model_emax_binary(model), 19
model_exp(model), 19
model_exp_binary(model), 19
model_independent(model), 19
model_independent_binary(model), 19
model_linear (model), 19
model_linear_binary (model), 19
model_loglinear (model), 19
model_loglinear_binary (model), 19
model_logquad (model), 19
model_logquad_binary (model), 19
model_longitudinal, 19, 31
model_longitudinal_idp
  (model_longitudinal), 31
model_longitudinal_itp
  (model_longitudinal), 31
model_longitudinal_linear
  (model_longitudinal), 31
model_quad (model), 19
model_quad_binary (model), 19
plot.dreamer(dreamerplot), 3
plot.dreamer_bma(dreamerplot), 3
plot_comparison, 33
plot_trace, 35
post_medx, 39
post_perc_effect, 42
posterior, 36
pr_eoi, 44
pr_med, 46
pr_medx, 47
summary.dreamer, 49
summary.dreamer_bma, 51